# CSCU9N5
# Multimedia and HCI
# Creating a Prototype Web-based Multimedia Application: An Intuitive Introduction to Metaheuristics

2830974

`Main HTML file: index.html`

## 1 Introduction

"An Intuitive Introduction to Metaheuristics" is an interactive instructional course that is available as a web-based multimedia application. It aims at undergraduate, postgraduate and doctoral students as well as teachers that study or teach computer science. It is especially suitable for those students that already have some experience with implementing algorithms but are new to the field of metaheuristics and optimization. The application's goal is to provide an intuitive understanding of the most important metaheuristic algorithms. It is also suitable for professors and teachers who are looking for new ways to teach their students complex concepts.

## 2 Product description and user personas

In this section, the application is described in more detail, the product delivery is presented, and two specific user personas are introduced.

### 2.1 The product

After giving a brief motivation on the use of metaheuristics, the application specifically introduces two classes of metaheuristic algorithms. Those algorithms that keep only a single solution during the optimization (single-point algorithms) and those that keep a population of solutions (population-based algorithms) [6]. Single-point algorithms that the user can access are: Hill-climbing, simulated annealing and tabu search. The available population-based algorithms are: Evolutions strategies, a genetic algorithm and the ant colony optimization algorithm. These six algorithms were chosen because they are often similar to each other and help in understanding further metaheuristic algorithms. For example, simulated annealing is similar to the hill-climbing algorithm in certain scenarios (when the temperature equals zero) [6]. To help the users understand quickly, the algorithms are shown with the help of interactive animations, videos and pictures. The intuitive design supports a fast navigation through the app to help the user build a consistent mental model [1, 9, 8]. To strengthen the user's understanding further, assignments for implementing the algorithms are provided. Teachers can use the assignments to evaluate their student's ability. Furthermore, teachers can request the assignments's solutions via the contact web form.

The application is delivered as a web-based application in a zip-folder. This delivery method has the advantage that the application can also be used offline. The internet is only required for sending requests via the contact form. The folder can be stored as an electronic representation on the university computers. The application can be started in any browser by opening the index.html file that can be easily found in the folder. It has a responsive design that scales to smaller browser windows and to smartphone screens. Therefore the application can be used in many situations. For example, for demonstrating an algorithm during a lecture on one of the university computers or for reading about an algorithm while commuting to the university.

### 2.2 The users

Two personas, one teacher and one student, are presented and described below. The personas' descriptions follow the structure of Benyon [1].

**Fig. 1.** Maria is a professor at the University of Scotland [4].

**Maria** is a 47-year-old professor at the University of Scotland where she is part of the computer science department (Figure 1). She teaches an introductory course on metaheuristics. She possesses a lot of knowledge on this topic and conducts research in this area. Because she always wants to improve her classes, she likes to try new ways of teaching. For example, she records her lectures so that students have the chance to listen to them again. Apart from her work, Maria has a husband and is mother of two children. In her free time she likes to go on trips to explore the Scottish Highlands with her family. She lives twenty minutes away from the university campus.

**Goals:** Ideally, Maria wants to inspire students to continue with her advanced course on metaheuristics once they completed her introductory course. She also likes to find talented students who are interested in joining her research group as part of the universities' doctoral program. Maria always tries to receive feedback from students about her course and her teaching methods. Since metaheuristics is a very broad topic her goal for the introductory course is to teach students the intuition about some important metaheuristic algorithms. Furthermore, she wants to show her students why metaheuristics is an important field for studies or research. At the end of the semester, Maria must evaluate the students that attended her course. For this reason, she usually provides some graded assignments. While Maria is used to teaching with textbooks, PowerPoint slides and research papers she is currently searching for a simple and intuitive web application that she can recommend to her students at the start of the semester. Ideally this application should be interactive, and it should include assignments in order to help her with the assessment of the students.

**Technology Skills:** Maria is very experienced in the use of multimedia to support her teaching. She knows how to use laptops and smartphones. Furthermore, she is proficient in the programming languages C, Java and Python.

**Limitations:** Due to her family responsibilities, she has limited time to prepare assignments.

**Fig. 2.** Daniel is an undergraduate student at the University of Córdoba, Spain [4].

**Daniel** is in his fourth year of undergraduate studies in computer science at the University of Córdoba, Spain (Figure 2). He is 22 years old and he enjoys learning about new concepts and algorithms. His older brother studied computer science and sparked Daniel's interest in the subject. In order to understand new algorithms, Daniel usually tries to understand the usefulness as well as the intuition behind them. This semester, Daniel attends a course on metaheuristics. He does most of his studying in the university's library using his laptop, especially when working on assignments. When he needs to read articles, he sometimes also uses his phone to study while he commutes to his flat which is 25 minutes away from the university campus.

**Goals:** When it comes to assignments and exams, Daniel's goal is always to perform very well. After graduation, he eventually wants to move on to pursue doctoral studies in computer science just like his older brother. This requires him not only to get good marks but also to develop a deep understanding for problems in computer science. To develop this kind of understanding, his goal is to implement every important algorithm from his lectures at least once.

**Technology Skills:** Daniel has extended knowledge about laptops and smartphones. He possesses solid Java and Python programming skills and has already implemented a good number of algorithms.

**Limitations:** While he speaks fluent English, Daniel sometimes struggles with difficult English words and grammar. He is not a native speaker and thus prefers simple English.

# 3 Product design

This section introduces the design documentation. Firstly, the concrete scenarios for Maria and Daniel are presented. Secondly, the wireframes of the application are shown. Thirdly, the navigation map is given and lastly, the design decisions are justified and supported with references.

## 3.1 Concrete scenarios

Two concrete scenarios are described in the following paragraphs. They build on the PACT analysis and follow the structure that is suggested by Benyon [1].

**Concrete scenario 1**

**Description:** Testing a new way of teaching
**People:** Maria
**Activities:** Teaching the hill-climbing algorithm
**Context:** Lecture on metaheuristics
**Technology:** A laptop and a projector

Maria (**people**) wants to introduce the hill-climbing algorithm (a single-state algorithm) in her lecture today (**context**). Because she really likes the interactive animation in the web application, she opens it on her laptop during the lecture and shows it to her students via the projector (**technology**). On the start screen, she clicks the button that takes her straight to the single-state algorithms. Then she clicks on the hill-climbing algorithm button. Maria explains the concept behind the algorithm and uses the interactive animation to visually show the concept (**activity**). The animation shows a ball (the solution) that is in a valley (the current solution) and that needs to get up on the hill (a better solution). Each time Maria clicks on the ball, it climbs up the hill a little further. The solution gets better. This is indicated by the color of the ball. It slowly turns from black to red. After her fourth click, the solution arrives up on the hill (local maximum). Maria's students seem to understand the concept. At the end of the lecture, she clicks on the hill-climbing assignment. The assignment is on the same page as the hill-climbing visualization. She tells her students that they will be assessed on this assignment in the next practical session.

**Concrete scenario 2**

**Description:** Working on an assignment
**People:** Daniel
**Activities:** Understanding an algorithm
**Context:** Course on metaheuristics
**Technology:** A Laptop and a smartphone

For his course on metaheuristics (**context**), Daniel (**people**) has received an assignment. He is supposed to implement and hold a presentation on one metaheuristic algorithm of his choice. After consulting the recommended textbook, he feels overwhelmed by the amount of math that the textbook presents. Fortunately, his professor also recommended a web application that gives an intuitive introduction to metaheuristics. As Daniel is currently on the bus to the library, he opens the application on his smartphone (**technology**). Because he only had one lecture so far, he does not really know what metaheuristics are. He clicks on the "What's that?" button on the start screen, reads through the motivation section and enjoys the simple English language. Then he clicks on the population-based algorithm section in the navigation bar. He goes on to click on the evolution strategies algorithm button. Daniel plays the animation and starts to understand the main idea of the algorithm (**activities**). He also gets an idea of how he will present the algorithm in class. After arriving at the library, he opens the evolution strategies page on his laptop (**technology**) and uses the provided pseudocode to implement evolution strategies. Afterwards he creates a PowerPoint presentation in which he references the evolution strategies animation of the web application.

### 3.2 Wireframes

The following wireframes give a blueprint to the application's major screen layouts. They show the different elements that are used and how these work together [9]. The screen layout for the single-state algorithms is not depicted, because it is similar to the screen layout of the population-based algorithms (Figure 4). The same applies to the six algorithm pages that are all similar in structure. They are represented by the evolution strategies wireframe (Figure 5).

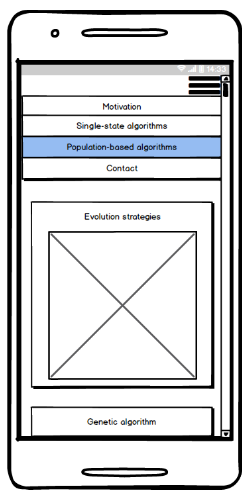


**Fig. 3.** Wireframe for the mobile screen. It shows the responsiveness of the design as well as the hamburger navigation menu.

Figure 3 shows the applications layout on a smartphone screen. The navigation bar turns into a hamburger menu that can be collapsed by the user. The buttons for the different algorithms are now presented vertically in order to make the design fit the smartphone screen.

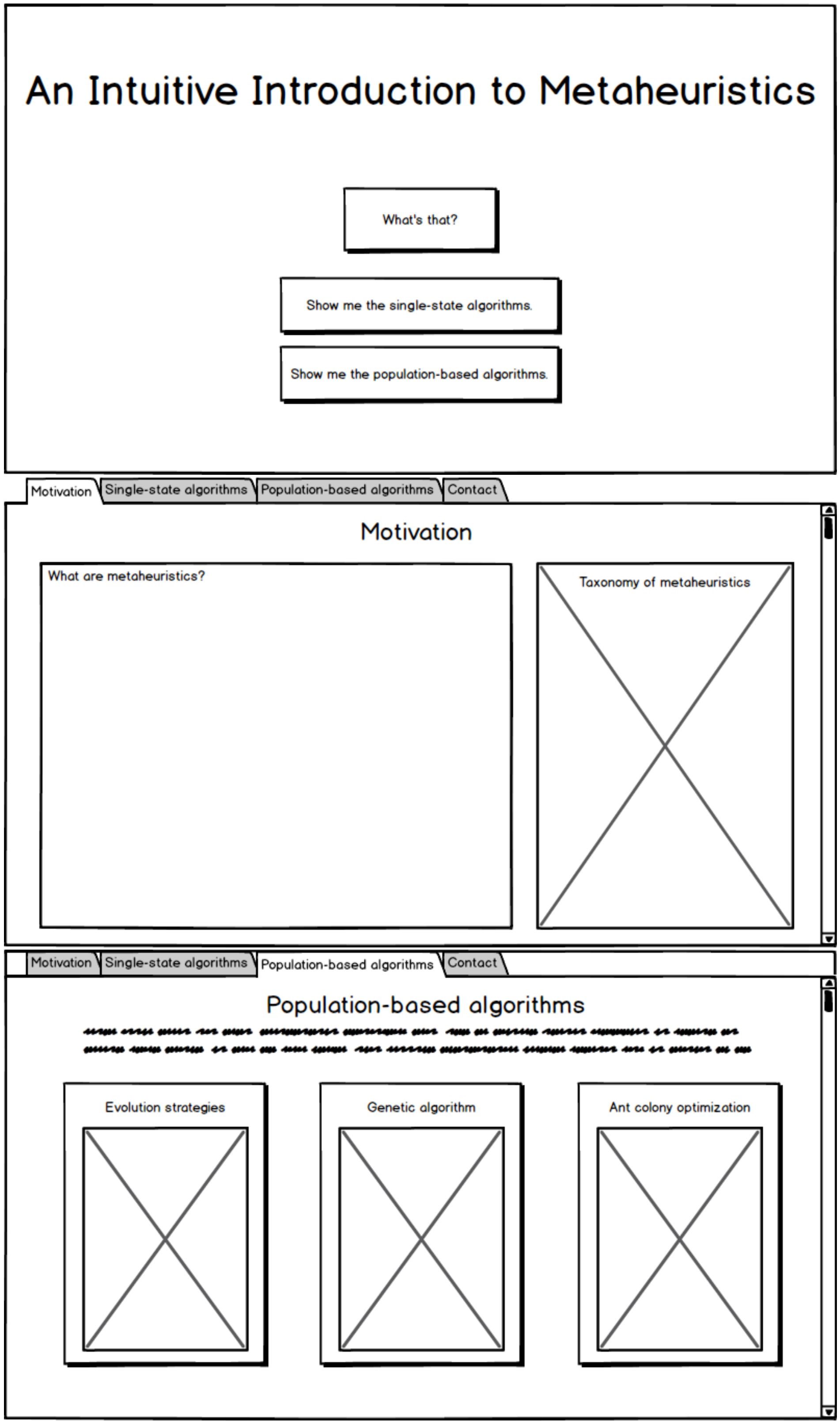


**Fig. 4.** Wireframes for the start screen (top), the motivation (middle) and the selection of the population-based algorithms (bottom).

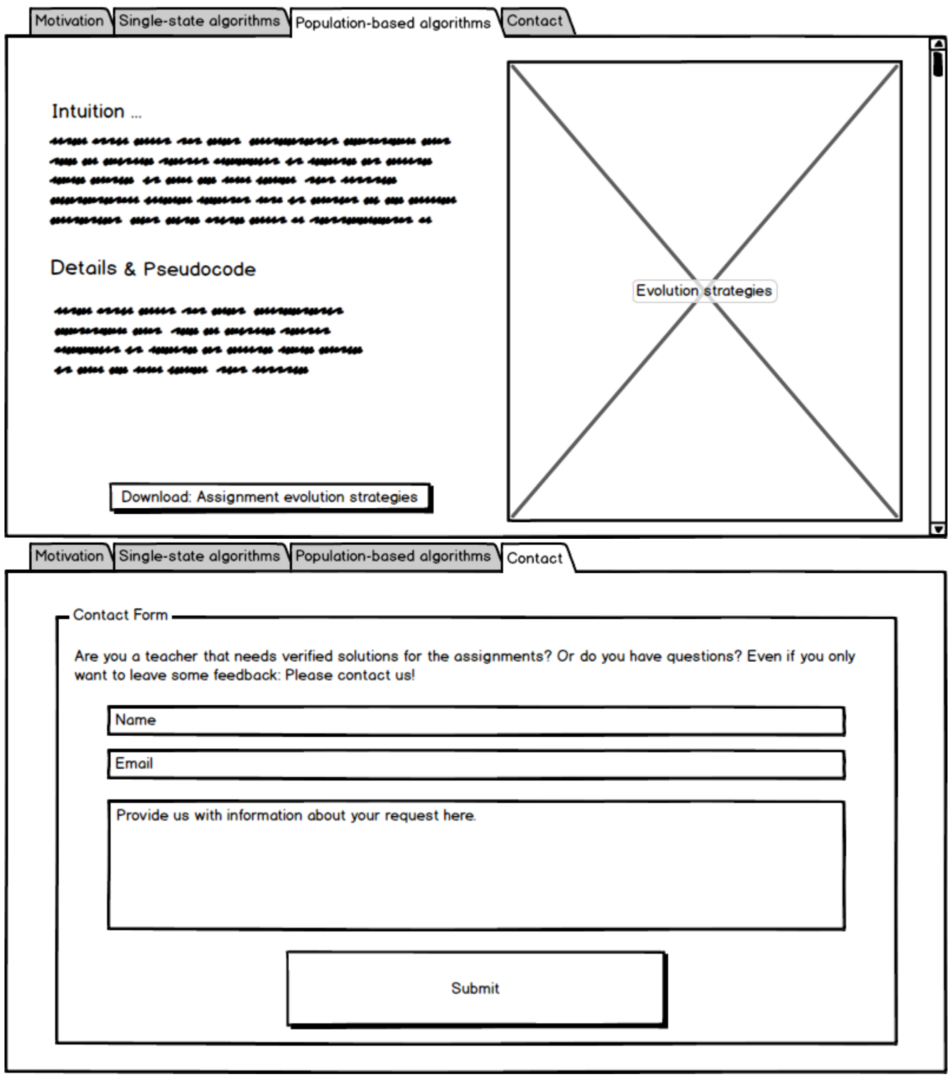


**Fig. 5.** Wireframes for the algorithm screen structure (top) and the contact form screen (bottom).

### 3.3 Navigation map

The navigation map is shown in Figure 6. When the application is opened, it shows the start screen. The users can access three of the four buttons of the navigation bar on the start screen. Once the users arrived at the navigation bar, they cannot go back to the start screen. This decision was made on purpose to separate user groups in the beginning (Section 3.4). The four pages that have buttons in the navigation bar can access each other and each can also access itself. From the single-state algorithms, from the population-based algorithms, and from the contact pages another hierarchy level can be accessed. The single-state and population-based algorithm pages have access to their respective algorithm pages. The four buttons of the navigation bar can be accessed from each algorithm page. On the contact page, a message can be submitted. Submitting a message, leads the user to the message submitted page. From the message submitted page all four navigation bar buttons can be reached.

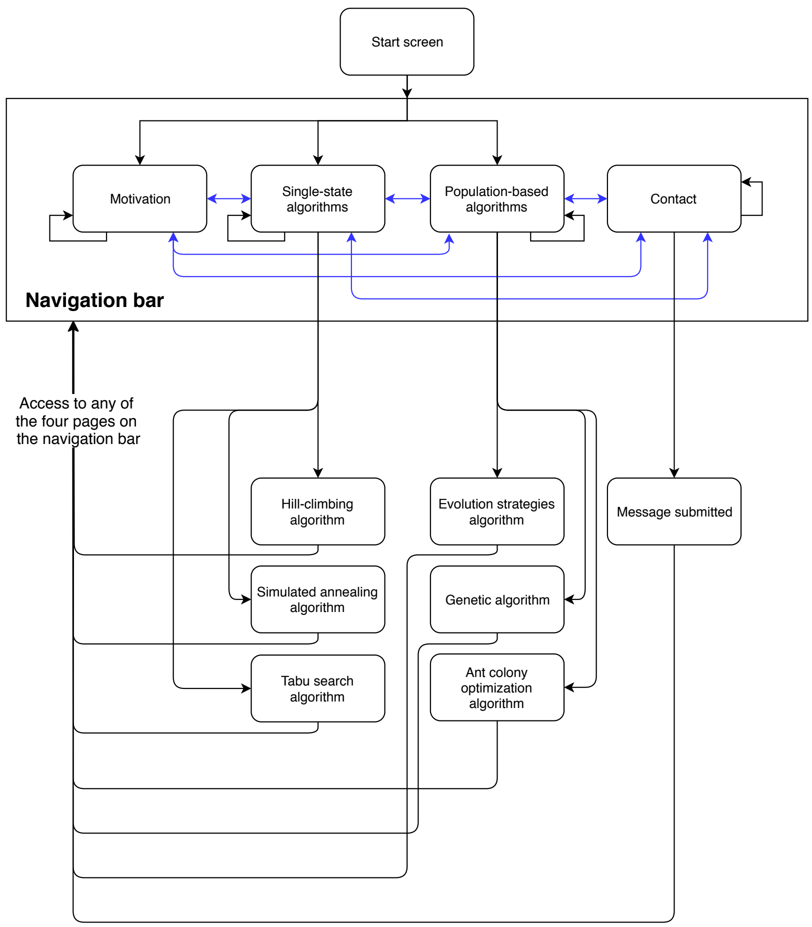


**Fig. 6.** The navigation map of "An Intuitive Introduction to Metaheuristics".

### 3.4 Design decisions

The application was designed with two goals in mind:

1) In order to achieve a great user experience design, the application should follow Shneiderman's three principles. That includes following most of his eight golden rules for interface design (Principle 2) [10].

2) It should put people first (PACT) [1] and therefore follow additional design principles that make the users' life easier.

In the following two paragraphs, the design decisions for achieving the first goal are explained. Then the design decisions for achieving the second goal are explained in the third paragraph.

Once the application is accessed, the start screen prompts the user with three different buttons. The "What's that?" button leads to the motivation section and the other two buttons lead to the single-state algorithms or population-based algorithms. This is done to separate between users that have never used the application and the ones who know the application already. The idea behind this design is to overcome the uncertainty of first-time users who have never heard about metaheuristics, while simultaneously providing buttons for intermediate or expert users to directly go to the algorithms [10]. This satisfies the first Shneiderman principle "recognize diversity" as well as one of his golden rules: "universal usability" [10]. After recognizing the diversity of the user, the start screen adds no additional value and hence there is no navigation icon that leads the user back to it (Principle 3: Error prevention). Each algorithm page consistently follows the same structure (intuition, details, pseudocode and assignment). Additionally, the application's navigation bar and footer are consistently colored in blue and present on every page (golden rule: consistency). Only on the start screen the navigation bar is substituted by the above-mentioned buttons. The color blue avoids problems with red-green color blindness and represents trust [2]. Trust is needed, because the user is supposed to download the assignments from the algorithm pages. The dark blue button in the navigation bar indicates the area of the application that the user is currently accessing. Once the user chooses a new section, it will be colored in dark blue, while the last visited section gets the brighter blue of the navigation bar again (Figure 7).


**Fig. 7.** An excerpt of the applications navigation bar.

In this way the application provides a feedback for each of the user's actions (golden rule: feedback [10]). Additionally, when hovering over the buttons on the navigation bar, the respective buttons are highlighted. This gives the user a visual cue of what to do with the interface (an affordance). The application also provides feedback in form of sound (e.g. after a click on the ball in the hill-climbing animation and after starting the evolution strategies animation). An example for the golden rule "closure" is the interactive hill-climbing animation. The user gets the hint that four clicks are needed to get the ball to the top of the hill. After the first click the ball moves a little bit up the hill (beginning). The user sees the ball getting closer to where it should be and knows simultaneously that more clicks are needed (middle). After the fourth click, the ball is on top of the hill. That provides the user with the visual feedback that a group of actions is completed and that the task is solved (end) [10]. Each button in the navigation bar has little icons to the left of it (Figure 7). They help the user to distinguish between single-state algorithms with one solution (one-person icon) and population-based algorithms with a population of solutions (three-person icon). The idea is that the icons reduce the user's "short-term memory load" (golden rule) [10]. When using the application the user "feels in control" (golden rule), because the application only changes according to the user's actions. Otherwise it does nothing. For example, when the user makes the browser window smaller, the application will adjust accordingly.

To achieve the second goal, several other design principles are followed as well. For example: On the single-state algorithm and population-based algorithm page, three big buttons with pictures for each algorithm are shown. This is done on purpose. According to "Hick's Law", every additional choice increases the time that users need in order to make decisions. Therefore, only three choices (buttons) are provided on each page. According to "Fitts's law" big buttons are easier to use because they can be reached quickly [5]. Most of the applications content (except for the six algorithm pages) can be accessed without scrolling ("above the fold") when viewed on a laptop with a 15-inch screen. This keeps the content relevant, as the content above the fold receives the most attention from the user [3]. All the applications dialogues try to use as little text as possible in order to be short and compact. This follows the "aesthetic and minimalist design" heuristic of the Nielsen Norman Group. [7]. On small devices the navigation bar can take up a lot of space on the screen. In order to give the user the opportunity to make the navigation bar smaller, a hamburger menu pops up when the screen width 810 pixels or smaller. The hamburger menu is a very recognizable metaphor and hence most users know what it means and how to use it [11]. Lastly, the application leaves a good amount of white space on its pages in order to not appear busy or cluttered [5]. The reason for this is that users prefer websites with less visual complexity [12].

## 4 Prototype description

The prototype version contains the major screen layouts as designed in the wireframes (Section 3.2) to demonstrate all essential functionalities of the full version. The navigation (as it is shown in the navigation map (Section 3.3)) is fully implemented in order to let the user experience the full scope of it. The prototype gives an example for all types of multimedia elements that will be employed in the full version: Interactive animations, sounds, images, and videos. However, it does not contain all algorithms. Tabu search, the genetic algorithm, and the ant colony optimization are not implemented. The pages of these algorithms would not test any new elements. Their page structure and their multimedia elements will be similar to those of the already implemented algorithms. As the functionality for sending requests via the contact form does not affect the design itself, it is not implemented in the prototype version.

## 5 Usability testing

The usability testing was carried out during the practical session.

# 6 Sources for media and external CSS and JavaScript libraries

This section references all sources that were used for building the application in terms of media, external CSS and JavaScript libraries. These sources are also mentioned by comments in the source code.

## 6.1 Media

**Content:**

– Sean Luke, 2013, Essentials of Metaheuristics, Lulu, second edition, available at: https://cs.gmu.edu/ sean/book/metaheuristics/Essentials.pdf
– Salimans, T., Ho, J., Chen, X., Sidor, S., Sutskever, I.: Evolution strategies as a scalable alternative to reinforcement learning. (2017)
– The taxonomy is based on: https://en.wikipedia.org/wiki/Metaheuristic/media /File:Metaheuristicsclassification.svg

**Images:**

– https://academo.org/demos/3d-surface-plotter/ ?expression=7*x*y%2Fe%5E(x%5E2%2By%5E2)xRange=-2%2C2yRange=-2%2C2resolution=25 using the function using function 7*x*y/e(x2 + y2) in y range (-2,2) and xrange (-2,2)
– https://en.wikipedia.org/wiki/File:Maxparaboloid.svg
– https://au.mathworks.com/help/optim/examples/tutorial-for-the-optimization-toolbox.html
– https://www.pinterest.co.uk/pin/812759063974168349/
– https://en.wikipedia.org/wiki/Rastriginfunction/media/File:Rastriginfunction.png
– https://www.vexels.com/png-svg/preview/159257/dna-chain-sketch
– https://de.wikipedia.org/wiki/Ameisenalgorithmus/media/Datei:Acoshortpath.svg

**Video:**

– The video was downloaded from: https://www.youtube.com/watch?v=SC5CX8drAtU

**Sound:**

– The click sound was taken from: http://soundbible.com/783-Click.html.

## 6.2 CSS

– Fonts: https://fonts.googleapis.com/css?family=Open+Sansdisplay=swap.
– Parts of the navigation bar were taken from: https://www.w3schools.com/css/cssnavbar.asp
– Parts of the button styling was taken from: https://www.youtube.com/watch?v=Y5SHm53WFEk
– Hover effect: http://www.corelangs.com/css/box/hover.html
– Parts for the side by side columns were taken from: https://www.w3schools.com/howto /howtocssimagessidebyside.asp
– Other parts for the side by side columns were taken from:https://www.w3schools.com/howto/tryit.asp ?filename=tryhowcsstwocolumnsresponsive
– Contact form: https://www.w3schools.com/howto/howtocsscontactform.asp.
– Parts of the responsive navigation were taken from: https://www.w3schools.com/howto /howtojstopnavresponsive.asp
– Hamburger menu with toggle functionality: https://www.youtube.com/watch?v=xMTs8tAapnQ
– Download button from: https://www.w3schools.com/howto/tryit.asp?filename =tryhowcssdownloadbutton

### 6.3 JavaScript

- Parts of the JavaScript code were taken from the CSCU9N5 Practical 2019, University of Stirling.

## References

1. Benyon, D.: Designing User Experience: A Guide to HCI, UX and Interaction Design. Pearson (2019), https://books.google.co.uk/books?id=dBcJMQAACAAJ
2. Blog, U.T.: How color impacts conversion rates and ux (2019), https://www.usertesting.com/blog/color-ux-conversion-rates/; Accessed: 10.11.2019
3. Fessenden, T.: Scrolling and attention (2018), https://www.nngroup.com/articles/scrolling-and-attention/; Accessed: 10.11.2019
4. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. arXiv (2018), https://thispersondoesnotexist.com/
5. Laja, P.: 8 web design principles to know in 2019 (2019), https://conversionxl.com/blog/universal-web-design-principles/; Accessed: 30.10.2019
6. Luke, S.: Essentials of Metaheuristics. Lulu, second edn. (2013), available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/
7. Nielsen, J.: 10 usability heuristics for user interface design (1994), https://www.nngroup.com/articles/ten-usability-heuristics/; Accessed: 30.10.2019
8. Norman, D.A.: The Design of Everyday Things. Basic Books, Inc., New York, NY, USA (2002)
9. Shankland, C., Graham, B.: Cscu9n5 lecture (2019)
10. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Publishing Company, USA, 5th edn. (2009)
11. Tran, T.H.: 10 pros and cons of the hamburger menu (with examples) (2019), https://www.invisionapp.com/inside-design/pros-and-cons-of-hamburger-menus/; Accessed: 30.10.2019
12. Tuch, A.N., Presslaber, E., Stoecklin, M., Opwis, K., Bargas-Avila, J.: The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments. International Journal of Human-Computer Studies pp. 794–811 (2012)